Robust Hashing for Image-based Malware Classification

Wei-Chung Huang, Fabio Di Troia and Mark Stamp
Department of Computer Science, San Jose State University, San Jose, California, U.S.A.

Keywords: Robust Hashing, Malware, Support Vector Machine, Wavelet, Distributed Source Coding.

Abstract: In this paper, we compare and contrast support vector machine (SVM) classifiers to robust hashing based strategies for the malware classification problem. For both the SVM and robust hashing approaches, we treat each executable file as a two-dimensional image. We experiment with two image-based robust hashing techniques, one that relies on wavelet analysis, and one that uses distributed coding. For our support vector machine experiments, we consider an image-based feature that deals with horizontal edges. While the SVM performs slightly better, there are some potential advantages to robust hashing for malware detection.

1 INTRODUCTION

Malware is software that is intentionally designed to cause harm to computer systems (Aycock, 2006). Due to our heavy reliance on computers in general, and software in particular, malware classification is a vitally important topic in information security. Signature scanning (i.e., pattern matching) is the most common form of malware detection, and hence malware writers have developed many concealment strategies aimed at defeating standard signature scanning techniques (Aycock, 2006). These concealment strategies result in malware families consisting of large numbers of related variants. Hence, any practical malware detection or classification strategy must be aimed at entire families, rather than individual malware samples.

Malware classification can be based on static analysis or dynamic analysis, or a combination of the two. Static malware analysis relies on features that can be extracted without executing (or emulating) the code—mnemonic opcodes are a well-known example of a static feature (Yewale and Singh, 2016; **?**). Dynamic malware analysis considers the behavior of software, which requires the software to be executed (or emulated) when extracting such features. API calls are a common dynamic feature (Alazab et al., 2010).

In this research, we treat malware samples as two-dimensional images and extract static features. We then apply techniques from image robust hashing (Venkatesan et al., 2000) and machine learning to analyze the samples based on these features. Our goal is to compare the effectiveness of a relatively straightforward and intuitive machine learning technique to robust hashing based strategies. The authors are not aware of any previous research that considers robust hashing techniques in the context of malware detection or analysis.

The remainder of this paper is organized as follows. In Section 2 we discuss relevant background topics, including treating malware as images, an introduction to robust hashing, the features we consider, and so on. In Section 3, we give our experimental results, and provide context for these results. Finally, Section 4 contains our conclusion and a brief discussion of possible future work.

2 BACKGROUND

In this section, we first motivate the work in this paper, by presenting examples of executable files viewed as images. Next, we give a brief overview of related previous work, followed by an introduction to robust hashing. We conclude this section with a discussion of the various image features we consider, and an outline of the machine learning technique that we use.

2.1 Executables as Images

We can treat any binary file as an image by simply interpreting the data as a two-dimensional array of values and converting to the desired image format. Intuitively, we might expect that some common malware obfuscation techniques can be mitigated by vie-

Huang, W-C., Troia, F. and Stamp, M.
Robust Hashing for Image-based Malware Classification.
DOI: 10.5220/0006942204510459
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS*, pages 451-459
ISBN: 978-989-758-319-3

wing an executable as an image. That this is indeed the case is strikingly illustrated in Figure 1, which includes images corresponding to three samples from each of three different malware families. For comparison, benign executables (as images) appear in Figure 2.

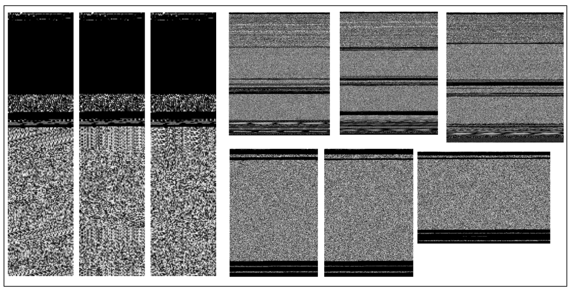
Figure 1: Images from malware families Agent.FYI (3 left-most), C2LOP.P (3 top right), and Alueron.gen!J (3 bottom right).

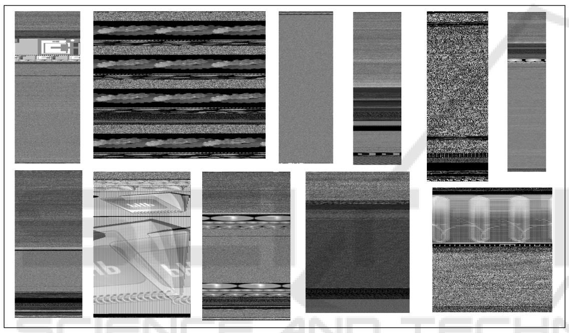
Figure 2: Examples of benign executables as images.

From the examples in Figures 1 and 2, it appears that image-based analysis has the potential to be a strong technique for discriminating between malware families, and possibly also for discriminating between malware and benign applications. Furthermore, we can easily deduce the structure of an executable from its image. For example, in Figure 3, we can clearly see the various segments that are present in this executable.

2.2 Previous Work

The paper (Nataraj et al., 2011) considers malware classification based on image analysis, and it is shown that malware samples from the same family tend to have similar image characteristics. In (Torralba et al., 2003), high level image features known as GIST descriptors are applied to the malware problem, and strong results are obtained. The work in (Yajamanam et al., 2018), builds on previous image-based malware work and obtains improved results using neural networks and transfer learning.

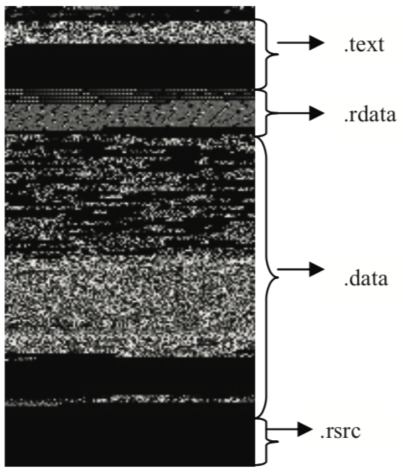

Figure 3: Code sections corresponding to a malware image.

The paper (Venkatesan et al., 2000) proposed robust image hashing to handle the proliferation of digital images. This robust hashing technique has many uses, including image indexing and image authentication (Lin and Chang, 1998; Schneider and Chang, 1996; Zhao et al., 2013). The robust image hashing method considered in (Zhao et al., 2013) uses both global and local features, while (Victor, 1994) presents a framework for the study of texture perception.

Feature extraction for robust image hashing is considered in (Monga et al., 2006), where the proposed process consists of feature vector extraction followed by compression. The technique in (Venkatesan et al., 2000) relies on wavelet decomposition for image feature extraction.

Another example of a novel robust image hashing technique can be found in the paper (Johnson and Ramchandran, 2003), where dithering and distributed source coding form the based for the technique. In this case, the syndrome serves as the hash value.

2.3 Robust Hashing

There are many types of hashing. For example, cryptographic hash functions have a wide range of uses in security-related applications, ranging from digital signatures to blockchain technology (Nakamoto, 2009). A crucial property of a cryptographic hash is collision resistance, in the sense that it is computationally infeasible to find distinct inputs that hash to the same value (Stamp, 2017).

Robust hashing is "robust" in the sense that similar objects are supposed to hash to the same (or similar) value. Note that this is essentially the opposite of the collision resistance property of a cryptographic hash. Robust image hashing has proven useful in biometric authentication (Sutcu et al., 2005), image watermark verification (Schneider and Chang, 1996), and image indexing (Venkatesan et al., 2000), for example.

Again, the goal of robust image hashing is to identify similar images, that is, similar images should hash to the same value. As proposed by (Monga et al., 2006), from a high level, the process consists of extracting relevant features, generating an intermediate "hash" value, then applying a compression step to generate the final hash value, as summarized in Figure 4. Note that the purpose of the compression step is to negate minor differences or remove noise so that similar intermediate hash values are clustered to the same group.

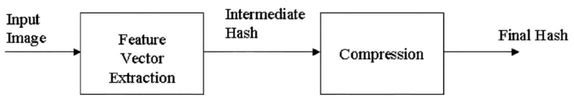

Figure 4: Process of robust hashing.

One approach that has been proposed for the compression step is the use of error-correcting codes (Venkatesan et al., 2000). Such codes have the desirable effect of ignoring small perturbations. A related approach can be found in (Johnson and Ramchandran, 2003), where a distributed source coding is considered. In this paper, we experiment with both of these techniques for the compression step. For the error-correcting code case, we use a simple Hamming code (Hamming, 1950).

Hamming codes are a class of linear error-correcting codes. Here, we use the Hamming(7, 4) code, a popular configuration with 4 data bits and 3 parity bits, which enables us to correct any 1 bit error.

Distributed source coding (DSC) reduces the computational burden of encoding (Pradhan and Ramchandran, 2003). Here, we employ a Wyner-Ziv encoder as developed and discussed in (Wyner and Ziv, 1976). A detailed discussion of DSC is beyond the scope of this paper; for additional information, see (Johnson and Ramchandran, 2003), for example.

2.4 Image Features

Image features can be categorized as either local or global. In this paper, we employ two local features and six global features. The local features we consider are the following.

Local Binary Pattern (LBP) is a well-known texture descriptor that is widely used in facial recognition (Ahonen et al., 2006). The LBP is found by extracting the local pixel contrast, relative to each pixel. The resulting histogram serves as the LBP feature vector.

Histogram of Oriented Gradient (HOG) is another texture descriptor that is wildly used due to its robustness (Dalal and Triggs, 2005). Whereas LBP is based on contrast, HOG is based on the local gradient.

In contrast to local features, global feature are focused on the entire image instead of local regions. The global features we consider are the following.

Horizontal Edges is a feature that has been used in many digital image processing applications. Based on the executable images in Figure 1, for example, it appears that a horizontal edge feature will be relevant for the malware classification problem considered here.

Specifically, we extract the distribution of horizontal edges from the edge magnitudes. Figure 5 illustrated this process. Note that we apply a low-pass filter, which serves to enhance the edges.

We then project the edge map magnitudes onto a one-dimensional space. Figure 6 illustrates the projection we obtain from the graph in Figure 7. We give a side-by-side comparison of the image and its horizontal edge projection in Figure 6.

Pixel Intensity is a useful global feature. This feature consists simply of a histogram of pixel intensities over the entire image.

Contrast is one of the most widely used features. The contrast tells us the variability in the brightness over an image.

Median Filter consists of low-pass filters with various noise reduction techniques applied. A median filter is designed to preserve edges while smoothing the image and hence this feature appears to be a good candidate for the malware image analysis problem.

Frequency Distribution is based on quantized discrete cosine transformation (DCT) coefficients. This feature measures the overall variability in the frequency domain.

Wavelet Transform can be used to generate features based on a series of different image resolutions (Daubechies, 1990). Roughly speaking, wavelet transforms enable us to analyze the signal in the frequency domain, while retaining temporal information. For our purposes, we decompose the image to several sub-bands as illustrated in Figure 8. We then compute statistical features from each band. For example, we compute the mean pixel value on the coarse band and the variance on the fine band.

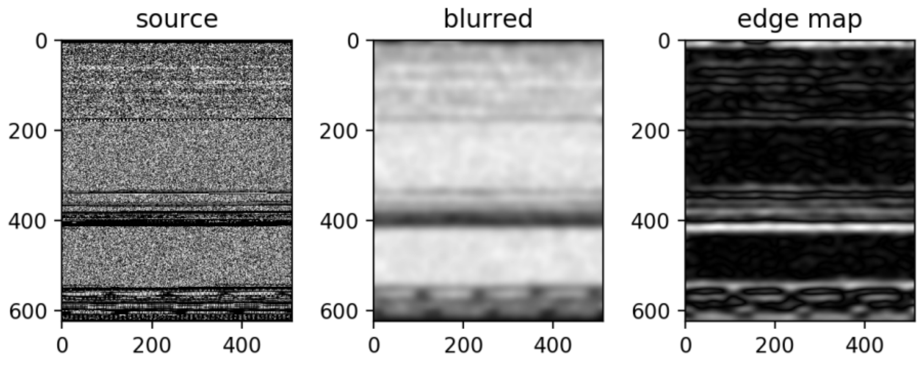

Figure 5: Horizontal edge feature extraction.

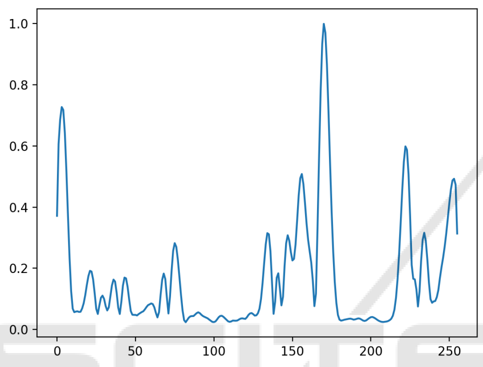

Figure 6: Horizontal edge magnitude feature vector.

2.5 Support Vector Machines

We compare our robust hashing technique to a machine learning classifier. For our machine learning experiments, we have chosen to use a support vector machine (SVM) (Cristianini and Shawe-Taylor, 2000). SVMs are well-known and widely used in many domains.

An SVM classifier attempts to determine a separating hyperplane where the margin (i.e., the minimum distance from the separating hyperplane to either class) is maximized. However, the data itself need not be linearly separable, in which case no separating hyperplane exists in the input space. In the SVM technique, we can shift the problem to a higher dimensional feature space by using a non-linear transformation via the so-called kernel trick. By moving to a higher dimension, there is a much greater chance that the training data will be linearly separable. The real beauty of SVMs is that we pay virtually no computational penalty for working in this higher dimensional feature space.

3 EXPERIMENTS

In this section, we give our experimental results. We apply three versions of robust hashing to the malware classification problem. For comparison, we also apply an SVM to the same malware classification problem. But first, we discuss our dataset.

3.1 Dataset

For the experiments in this paper, we use the Malimg dataset (Nataraj et al., 2011). As can be seen in Table 1, this dataset consists of 9,342 grayscale images representing binaries from 25 malware families of a variety of different types. Figure 9 gives examples of images found in this dataset.

Here, we deal with the problem of classifying the Malimg images into their 25 families. First, we consider an SVM, then we apply robust hashing techniques.

3.1.1 SVM for Classification

We extracted the features discussed in Section 2.4 for each of the malware images in the Malimg dataset. For all experiments discussed here, we use five-fold cross-validation, that is, we randomly partition the samples into five equal sized sets, S_1, S_2, S_3, S_4, S_5. In the first "fold," we train on sets S_2, S_3, S_4, S_5, reserving S_1 for testing; in the second fold, we train on S_1, S_3, S_4, S_5, reserving S_2 for testing, and so on. Cross validation serves to minimize any bias in the data, while also maximizing the number of test cases.

Figure 10 gives our results for the classification problem using an SVM based on the edge feature, as discussed in Section 2.4. In this case, we obtain an overall classification accuracy of about 84% if we consider the per family average. On the other hand, the classification accuracy is over 93% on a per sample basis, indicating that some of the less numerous

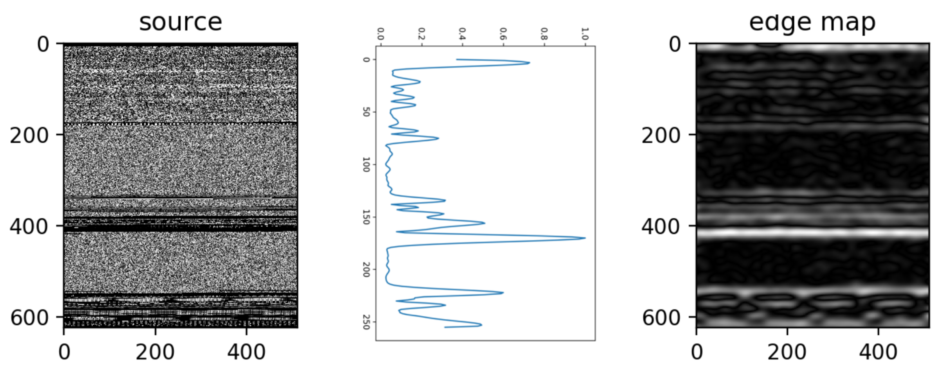

Figure 7: Horizontal edge feature vector and image.

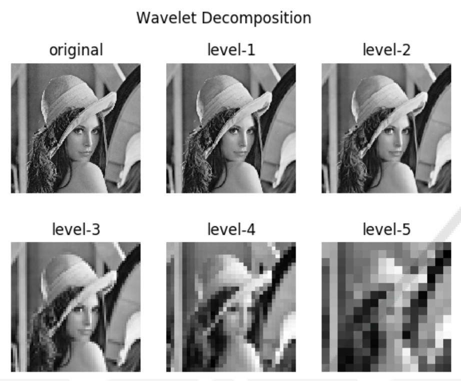

Figure 8: Sub-bands of wavelet decomposed image.

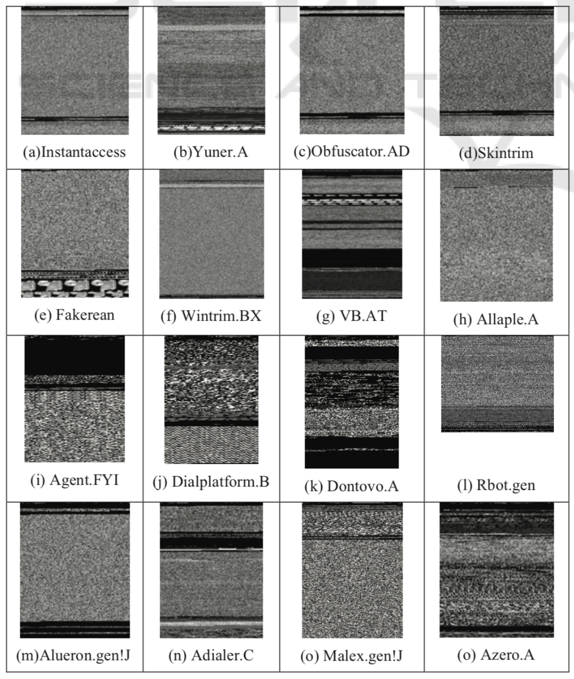

Figure 9: Malimg examples.

families are among those that are most difficult to distinguish. Furthermore, these results show that only a relatively small number of families are not classified

Table 1: Malimg Dataset.

	Family	Type	Samples
1	`Adialer.C`	Dialer	125
2	`Agent.FYI`	Backdoor	116
3	`Allaple.A`	Worm	2949
4	`Allaple.L`	Worm	1591
5	`Alueron.gen!J`	Trojan	198
6	`Autorun.K`	Worm	106
7	`C2LOP.gen!G`	Trojan	200
8	`C2LOP.P`	Trojan	146
9	`Dialplatform.B`	Dialer	177
10	`Dontovo.A`	Trojan DL	162
11	`Fakerean`	Rogue	381
12	`Instantaccess`	Dialer	431
13	`Lolyda.AA 1`	PWS	213
14	`Lolyda.AA 2`	PWS	184
15	`Lolyda.AA 3`	PWS	123
16	`Lolyda.AT`	PWS	159
17	`Malex.gen!J`	Trojan	136
18	`Obfuscator.AD`	Trojan DL	142
19	`Rbot!gen`	Backdoor	158
20	`Skintrim.N`	Trojan	80
21	`Swizzor.gen!E`	Trojan DL	128
22	`Swizzor.gen!I`	Trojan DL	132
23	`VB.AT`	Worm	408
24	`Wintrim.BX`	Trojan DL	97
25	`Yuner.A`	Worm	800

with high accuracy. For example, if we remove the seven families with the lowest accuracy from the set of 25 families, then we obtain an accuracy in excess of 98%. Overall, these results show that a simple and intuitive horizontal edge feature is extremely useful for this image-based malware classification problem.

Having determined appropriate parameters for the features, we now consider the problem of feature analysis. Specifically, we consider the tradeoff between the features used and accuracy. We apply both univariate feature selection (UFS) and recursive feature elimination (RFE). UFS simply consists of testing each feature independently and ranking them based on the

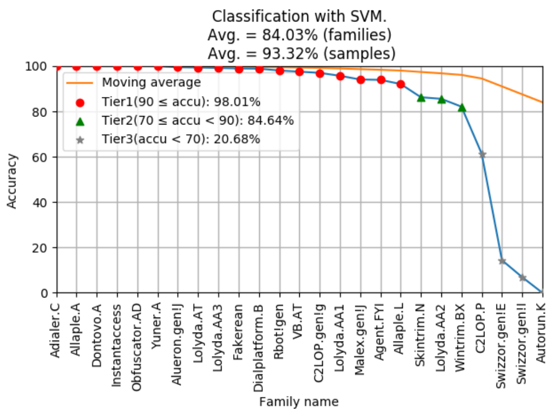

Figure 10: Classification accuracy using SVM.

accuracy of the resulting models. In RFE, we remove the lowest performing feature, based on linear SVM weights, then retrain the SVM on the reduced feature set. We generally expect RFE to be somewhat more reliable, as it accounts for the interactions between features.

Figure 11 depicts our UFS results for our global image features in the form of a bar graph. Perhaps not surprisingly, the horizontal edge feature outperforms all other features.

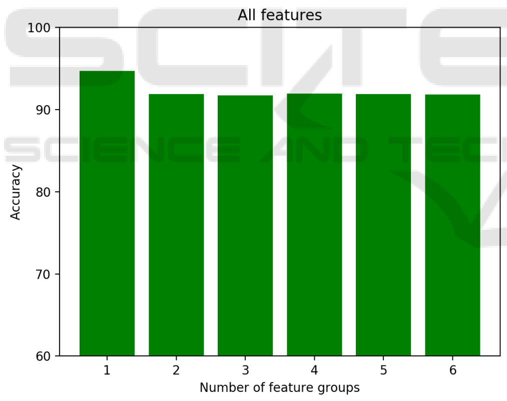

Figure 11: UFS results based on accuracy.

We further analyze the horizontal edge feature using RFE. Recall that this edge feature is 256 dimensional. When we apply RFE to this specific feature, we obtain the results in Figure 12. In this case, we find that by using 169 of the 256 dimensions, we can obtain optimal accuracy. We use this 169 dimensional set of horizontal edge features in the subsequent SVM experiments.

Using an SVM and the reduced horizontal edge feature as discussed above, we have achieved a classification accuracy of 92% for the 25 families in the Malimg dataset. Figure 13 gives confusion matrix (in the form of a heatmap) for this experiment. Note that in this figure, the families are numbered as in Table 1.

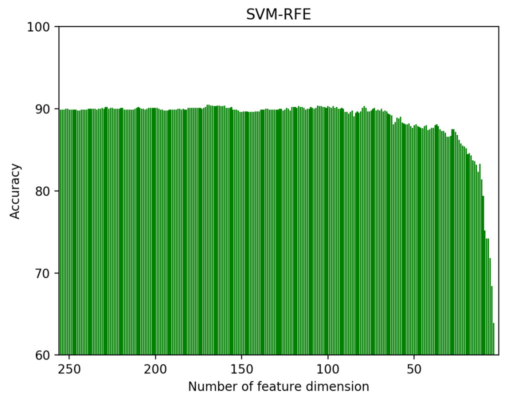

Figure 12: RFE result on horizontal edge feature.

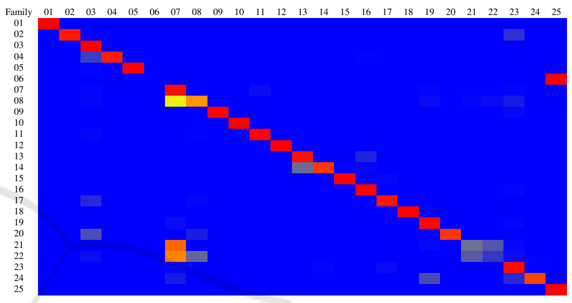

Figure 13: Confusion matrix for SVM as a heatmap.

Next, we apply two different robust hashing approaches to the malware classification problem. Recall that the robust hashing problem consists of feature vector extraction, followed by a compression step. In this research, we implement and test the image robust hashing technique from (Venkatesan et al., 2000), and the technique from (Johnson and Ramchandran, 2003), as well as a combined model.

3.1.2 Robust Hashing based on Wavelets

The robust hashing technique that is given in (Venkatesan et al., 2000) uses wavelet based analysis and an error-correcting decoder. In our implementation, we extract statistical information from different frequency bands and group together similar feature vectors based on an error-correcting decoder. Specifically, we use a 2-dimensional Haar wavelet transform and a five level decomposition. We then use a scalar quantization to normalize the feature vector over 128 levels. After all of these wavelet based features are acquired, we use a Hamming(7, 4) decoder to “correct” the sequence of quantized 7-bits values, thus generating a vector that represents an image family.

Classification results using this approach are summarized in Figure 14, where we obtain an overall accuracy of 79%. Although this is significantly lower than the 92% accuracy we obtained with an SVM, if we limit the analysis to the top 19 of the 25 families, we have a comparable accuracy of 90%, while with 11 families we can attain well over 95% accuracy.

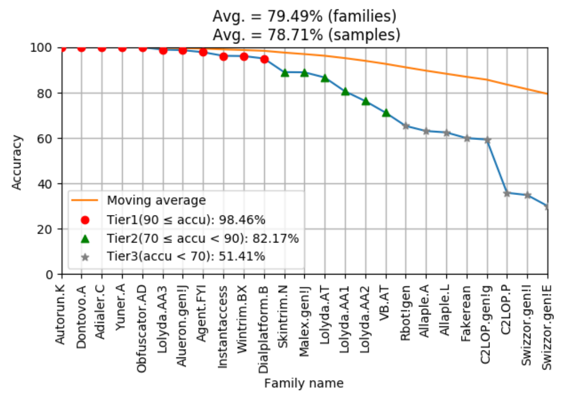

Figure 14: Classification result for wavelet approach.

3.1.3 Robust Hashing based on Distributed Coding

Next, following the general approach considered in the paper (Johnson and Ramchandran, 2003), we apply a distributed coding technique to the robust hashing problem. As implemented in the research (Varodayan et al., 2007), we have employed a Wyner-Ziv encoder based on DCT compression, and we use a Hamming code to produce a syndrome. The syndrome serves as the final robust hash value.

For this approach, we achieve slightly over 83% classification accuracy for all 25 families. Recall that the wavelet based approach was only able to attain about 79% accuracy. This is a significant improvement, but still not on par with the SVM results which achieved an accuracy of more than 92%.

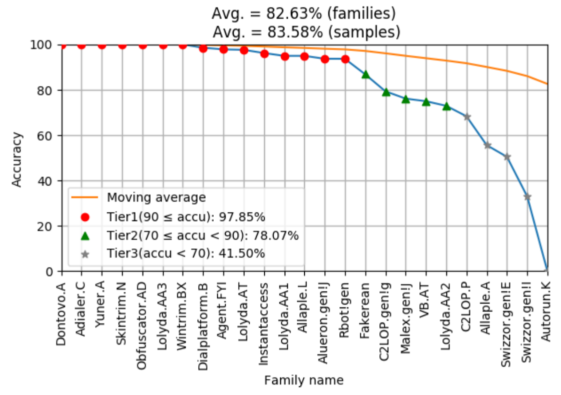

Figure 15: Classification result for distributed coding approach.

3.1.4 A Mulitphase Approach to Robust Hashing

From the robust hashing classification results above, we observe that for most families, one approach is more accurate than the other. For example, the family Autorun.k is classified well by the wavelet based classifier, but poorly by the distributed coding based classifier. Here, we combine both our wavelet based and our distributed coding based robust hashes—we refer to the resulting combination as a multiphase robust hash.

Intuitively, we can combine our two robust hashing techniques by scoring with both techniques, and in cases where the classifications disagree, we select the classification with the higher accuracy for its model. Figure 16 gives our results for this multiphase approach. Note that we achieve an accuracy above 87% over all 25 families, and we can attain an accuracy above 90% by discarding only one family. While this is not quite as strong as the 92% accuracy that we achieved with an SVM, it is comparable.

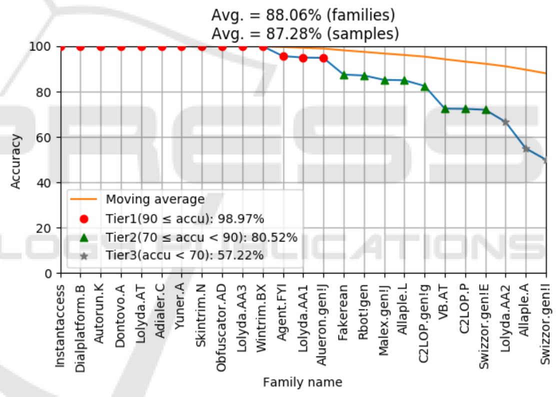

Figure 16: Classification result for multiphase approach.

Finally, Figure 17 compares the classification accuracy for the three robust hashing techniques considered. As with all of our classification experiments, these results are based on the 25 families in the Malimg dataset.

4 CONCLUSION

In this paper, we considered the problem of classifying images corresponding to 25 malware families. We compared a straightforward SVM classifier based on a simple and intuitive horizontal edge feature to robust hashing techniques. For the SVM, we carefully considered feature analysis issues, while for the robust hashing case, we experimented with three approaches. Overall, an optimized SVM performed so-

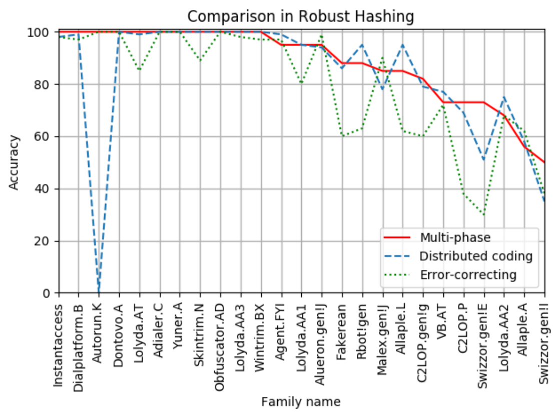

Figure 17: Comparison of robust hashing approaches.

mewhat better, but our best robust hashing results are comparable. With respect to robust hashing, we find that these techniques perform better than the SVM on most families, but a few malware families are extremely difficult to classify. In contrast, the SVM yields more consistent results over the 25 families.

In spite of the slightly lower accuracy, there are some potential advantages to robust hashing. Perhaps the biggest of these advantages is that in robust hashing, it is easy to add new families as they appear—we simple generate a syndrome (which, in effect, defines a cluster) for the new family and the remainder of the robust hashing classification model is unchanged. In contrast, for an SVM, we would need to retrain the model each time a new family is added. For large problems, the SVM training cost—in terms of both time and computational resources—would be substantial.

It appears that malware analysis is a novel application of robust hashing. And it is worth noting that robust hashing is a fairly general and somewhat amorphous concept. Hence, a multitude of variations on robust hashing can be considered. In addition to testing some of the many different possible forms or robust hashing, future work could include an analysis of additional features. We could also obtain a more fine-grained view by analyzing each section of an executable (i.e., `.text`, `.rdata`, and so on) separately. In addition, hybrid classification techniques involving robust hashing and any of a variety of machine learning techniques would be an interesting topic for further research. For example, we could apply various robust hashing techniques to a variety of image features, then apply a machine learning classifier to the results of these robust hashing algorithms.

Within the robust hashing paradigm, it would be worthwhile to experiment with more sophisticated coding techniques, such as Reed-Muller codes or trellis-coded modulation (TCM). Another type of hybrid model would consist of using non-coding based compression strategies within a robust hashing scheme—techniques such as K-means, EM clustering, k-nearest neighbor, and Gaussian mixture models, for example, would fit naturally within a robust hashing framework.

REFERENCES

Ahonen, T., Hadid, A., and Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041.

Alazab, M., Venkataraman, S., and Watters, P. (2010). Towards understanding malware behaviour by the extraction of api calls. In *Proceedings of the 2010 Second Cybercrime and Trustworthy Computing Workshop*, CTC '10, pages 52–59. IEEE Computer Society.

Aycock, J. (2006). *Computer Viruses and Malware*. Springer.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2005, pages 886–893. IEEE.

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005.

Hamming, R. W. (1950). Error detecting and error correcting codes. *Bell Labs Technical Journal*, 29(2):147–160.

Johnson, M. and Ramchandran, K. (2003). Dither-based secure image hashing using distributed coding. In *Proceedings of 2003 International Conference on Image Processing*, ICIP 2003, pages 751–754. IEEE.

Lin, C.-Y. and Chang, S.-F. (1998). Generating robust digital signature for image/video authentication. In *Multimedia and Security Workshop at ACM Multimedia*, pages 49–54.

Monga, V., Banerjee, A., and Evans, B. L. (2006). A clustering based approach to perceptual image hashing. *IEEE Transactions on Information Forensics and Security*, 1(1):68–79.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. (2011). Malware images: Visualization and automatic classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, VizSec'11, pages 4:1–4:7. ACM.

Pradhan, S. S. and Ramchandran, K. (2003). Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643.

Santos, I., Brezo, F., Nieves, J., Penya, Y. K., Sanz, B., Laorden, C., and Bringas, P. G. (2010). Idea: Opcode-sequence-based malware detection. In *International Symposium on Engineering Secure Software and Systems*, pages 35–43. Springer.

Schneider, M. and Chang, S.-F. (1996). A robust content based digital signature for image authentication. In *Proceedings of 1996 International Conference on Image Processing*, ICIP 1996, pages 227–230. IEEE.

Stamp, M. (2017). *Introduction to Machine Learning with Applications in Information Security*. Chapman and Hall/CRC, Boca Raton.

Sutcu, Y., Sencar, H. T., and Memon, N. (2005). A secure biometric authentication scheme based on robust hashing. In *Proceedings of the 7th Workshop on Multimedia and Security*, pages 111–116. ACM.

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. https://www.cs.ubc.ca/ murphyk/Papers/iccv03.pdf.

Varodayan, D., Lin, Y.-C., Mavlankar, A., Flierl, M., and Girod, B. (2007). Wyner-ziv coding of stereo images with unsupervised learning of disparity. In *Proceedings of Picture Coding Symposium*.

Venkatesan, R., Koon, S.-M., Jakubowski, M. H., and Moulin, P. (2000). Robust image hashing. In *Proceedings of 2000 International Conference on Image Processing*, volume 3 of *ICIP 2000*, pages 664–666. IEEE.

Victor, J. D. (1994). Images, statistics, and textures: Implications of triple correlation uniqueness for texture statistics and the julesz conjecture: Comment. *Journal of the Optical Society of America A*, 11(5):1680–1684.

Wyner, A. and Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10.

Yajamanam, S., Selvin, V. R. S., Troia, F. D., and Stamp, M. (2018). Deep learning versus gist descriptors for image-based malware classification. In *Proceedings of the 2nd International Workshop on Formal Methods for Security Engineering*, ForSE 2018.

Yewale, A. and Singh, M. (2016). Malware detection based on opcode frequency. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2016 International Conference on*, pages 646–649. IEEE.

Zhao, Y., Wang, S., Zhang, X., and Yao, H. (2013). Robust hashing for image authentication using zernike moments and local features. *IEEE Transactions on Information Forensics and Security*, 8(1):55–63.